

APPARATUS AND METHOD FOR PROVIDING CONFERENCE CALL ROSTER INFORMATION WITH SPEAKER VOICE IDENTIFICATION

5

FIELD OF INVENTION

The present invention generally relates to identification of parties in a telephone conversation and specifically to an apparatus and method for identifying and indicating the speaker during a multiple party telephone conversation.

10 CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to the following concurrently filed applications: "Apparatus and Method for Transmission and Receipt of Conference Call Roster Information via the Internet" and "Apparatus and Method for Transmission and Receipt of Conference Call Roster Information via a Telephone Display Unit."

15 BACKGROUND OF THE INVENTION

Caller Identification (Caller ID) enables telephone conference caller participants to identify a caller before picking up the telephone. Caller ID is useful when the caller does not identify himself and immediately begins a conversation. Caller ID has solved the problem of awkwardly interrupting a telephone conversation in order to identify the other party involved.

20 However, Caller ID cannot identify the speaking party where there are multiple party telephone conversations, such as during a conference call.

Due to increased travel costs and the cost of office space, conference call meetings have increased in popularity. Frequently, the participants call into a conference calling service which links the appropriate callers together. To attend a conference call, the parties call a specific

telephone number and identify themselves with a conference call number specific to each party. All of the parties are then linked together on a common line and the conference call proceeds. During the conference call, it is often difficult to identify the speaker at any given time. Interrupting the conference call to identify the speaker is not preferable as it disrupts the flow of the conference call. Therefore, a need exists for an apparatus and a method for identifying and indicating the speaker during a conference call.

SUMMARY OF INVENTION

The present invention allows a participant in a conference call to receive conference call information at a telephone display unit, at a computer display or at both types of display. The conference call information is updated during the call and includes identification of the speaker during a conference call. The conference callers are connected together and identified either by standard caller identification or by other identification provided by the operator. The conference callers' identifications are then transmitted in two ways. First, a coded signal is sent over the telephone connection to a telephone display unit. Second, an information packet is transmitted over the Internet to a conference call participant's computer. A server computer monitors the conference call and the identification of parties involved is updated as parties are added or disconnected. The speaking party is identified in one two ways. First, during the call, the speaker's telephone line will have a higher transmission amplitude than the other telephone lines and the line with the highest transmission amplitude will be identified by the server computer. Second, voice identification data is gathered before the conference call and stored in the server computer for use in identifying the speaker.

BRIEF DESCRIPTION OF DRAWINGS

Figure 1 is depiction of a distributed data processing system;

Figure 2 is a depiction of a server computer;

5 Figure 3 is a depiction of a client computer;

Figure 4 is a depiction of the server computer connected to the telephone switchboard and to the Internet;

Figure 5 is depiction of the server computer attached to a modulator for sending a coded signal through the telephone connection;

10 Figure 6A is a depiction of the sever computer memory;

Figure 6B is a depiction of the organization of the data files in the computer memory;

Figure 6C is a depiction of the organization of the programs in the computer memory;

Figure 6D is a depiction of the conference call list file;

Figure 6E is a depiction of a plurality of conference call information files;

15 Figure 6F is a depiction of a incoming line list file;

Figure 6G is a depiction of a plurality of line information files;

Figure 6H is a depiction of a participant list file;

Figure 6I is a depiction of a plurality of participant information files;

Figure 6J is a depiction of a plurality of conference call line list files;

20 Figure 6K is a depiction of a plurality of conference call links files;

Figure 7 is a depiction of the telephone display unit;

Figure 8 is a depiction of the computer display;

10001752-103101
FOI/EO 7.5/FOUO

Figure 9A is a depiction of the information shown at the telephone display unit;

Figure 9B is a depiction of alternate information shown at the telephone display unit;

Figure 10 is a flow chart of the information packet program;

Figure 11 is a flow chart of the telephone display unit program;

5 Figure 12 is a flow chart of the voice amplitude acquisition program;

Figure 13 is a flow chart of the voice amplitude display program;

Figure 14 is a flow chart of the voice print acquisition program; and

Figure 15 is a flow chart of the voice print identification program.

10 DESCRIPTION OF PREFERRED EMBODIMENTS

Figure 1 depicts a pictorial representation of a distributed data processing system in which the present invention may be implemented and is intended as an example, and not as an architectural limitation, for the processes of the present invention. Distributed data processing system **100** is a network of computers which contains a network **102**, which is the medium used to provide communication links between the various devices and computers connected together within distributed data processing system **100**. Network **102** may include permanent connections, such as wire or fiber optic cables, or temporary connections made through telephone connections. In the depicted example, a server **104** is connected to network **102** along with storage unit **106**. In addition, clients **108**, **110**, and **112** also are connected to a network **102**. Clients **108**, **110**, and **112** may be, for example, personal computers or network computers.

For purposes of this application, a network computer is any computer, coupled to a network, which receives a program or other application from another computer coupled to the

network. In the depicted example, server **104** provides Web based applications to clients **108**, **110**, and **112**. Clients **108**, **110**, and **112** are clients to server **104**. Distributed data processing system **100** may include additional servers, clients, and other devices not shown. In the depicted example, distributed data processing system **100** is the Internet with network **102** representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another. Distributed data processing system **100** may also be implemented as a number of different types of networks, such as, an intranet, a local area network (LAN), or a wide area network (WAN).

Referring to Figure 2, a block diagram depicts data processing system **200**, which may be implemented as a server, such as server **104** in Figure 1 in accordance with the present invention. Data processing system **200** may be a symmetric multiprocessor (SMP) system including a plurality of processors such as first processor **202** and second processor **204** connected to system bus **206**. Alternatively, a single processor system may be employed. Also connected to system bus **206** is memory controller/cache **208**, which provides an interface to local memory **209**. I/O bus bridge **210** is connected to system bus **206** and provides an interface to I/O bus **212**. Memory controller/cache **208** and I/O bus bridge **210** may be integrated as depicted. Peripheral component interconnect (PCI) bus bridge **214** connected to I/O bus **212** provides an interface to first PCI local bus **216**. Modem **218** may be connected to first PCI bus local **216**. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers **108**, **110** and **112** in Figure 1 may be provided through modem **218** and network adapter **220** connected to first PCI local bus **216** through add-in boards. Additional PCI bus bridges such as second PCI bus bridge **222** and third PCI bus

bridge 224 provide interfaces for additional PCI local buses such as second PCI local bus 226 and third PCI local bus 228, from which additional modems or network adapters may be supported. In this manner, data processing system 200 allows connections to multiple network computers. A memory-mapped graphics adapter 230 and hard disk 232 may also be connected to I/O bus 212 as depicted, either directly or indirectly. Those of ordinary skill in the art will appreciate that the hardware depicted in Figure 2 may vary. For example, other peripheral devices, such as an optical disk drive and the like also may be used in addition or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention. The data processing system depicted in Figure 2 may be, for example, an IBM RISC/System 6000 system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system.

With reference now to Figure 3, a block diagram illustrates a data processing system in which the invention may be implemented. Data processing system 300 is an example of either a stand-alone computer, if not connected to distributed data processing system 100, or a client computer, if connected to distributed data processing system 100. Data processing system 300 employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Micro Channel and ISA may be used. Processor 302 and main memory 304 are connected to PCI local bus 306 through PCI bridge 303. PCI bridge 303 also may include an integrated memory controller and cache memory for Processor 302. Additional connections to PCI local bus 306 may be made through direct component interconnection or through add-in boards. In the depicted example, local area

network (LAN) adapter **310**, SCSI host bus adapter **312**, and expansion bus interface **314** are connected to PCI local bus **306** by direct component connection. In contrast, audio adapter **316**, graphics adapter **318**, and audio/video (A/V) adapter **319** are connected to PCI local bus **306** by add-in boards inserted into expansion slots. Expansion bus interface **314** provides a connection for a keyboard and mouse adapter **320**, modem **322**, and additional memory **324**. SCSI host bus adapter **312** provides a connection for hard disk drive **326**, tape drive **328**, and CD-ROM **330** in the depicted example. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors. An operating system runs on processor **302** and is used to coordinate and provide control of various components within data processing system **300** in Figure 3. The operating system may be a commercially available operating system such as OS/2, which is available from International Business Machines Corporation. "OS/2" is a trademark of International Business Machines Corporation. An object oriented programming system, such as Java, may run in conjunction with the operating system and provides calls to the operating system from Java programs or applications executing on data processing system **300**. "Java" is a trademark of Sun Microsystems, Incorporated. Instructions for the operating system, the object-oriented operating system, and applications or programs may be located on storage devices, such as hard disk drive **326**, and they may be loaded into main memory **304** for execution by processor **302**.

Those of ordinary skill in the art will appreciate that the hardware in Figure 3 may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash ROM (or equivalent nonvolatile memory) or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in Figure 3. Also, the processes of the present

invention may be applied to a multiprocessor data processing system. For example, data processing system 300, if configured as a network computer, may not include SCSI host bus adapter 312, hard disk drive 326, tape drive 328, and CD-ROM 330, as noted by dotted line 332 in Figure 3 denoting optional inclusion. In that case, the computer, to be properly called a client

5 computer, must include some type of network communication interface, such as LAN adapter 310, modem 322, or the like. As another example, data processing system 300 may be configured to be bootable without relying on some type of network communication interface. As a further example, data processing system 300 may be a Personal Digital Assistant (PDA) device which is configured with ROM and/or flash ROM in order to provide non-volatile memory for

10 storing operating system files and/or user-generated data. The depicted example in Figure 3 and above-described examples are not meant to imply architectural limitations with respect to the present invention. It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in a form of

15 a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such a floppy disc, a hard disk drive, a RAM, and CD-ROMs, and transmission-type media, such as digital and analog communications links.

20 Figure 4 is a depiction of Caller ID System 1(CIDS1) 400. CIDS1 has computer 410 connected to amplitude discriminator 420 by line 414 and tonetwork 102 by line 404. Computer 410 may be a server computer such as data processing system 200 in Fig. 2. In the preferred

embodiment network **102** is the Internet. Amplitude discriminator **420** is connected to switch **430** by line **424**. Switch **430** connects incoming telephone lines (TL) TL1 **432**, TL2 **436**, TL3 **440**, TL4 **444**, TL5 **448** and TL6 **452**. The incoming telephone lines may be wire or wireless and extend from public switch telephone network (PSTN) **480**. Switch **430** can configure the incoming lines to form any combination of conference calls. For purposes of illustration only, and not by way of limitation, conference call A (CCA) **460** and conference call B (CCB) **470** are shown. CCA **460** has TL1 **432**, TL3 **440** and TL4 **444** connected. CCB has TL2 **436**, TL5 **448** and TL6 **452** connected. Amplitude discriminator **420** determines the amplitude of the signal on TL1 **432**, TL2 **436**, TL3 **440**, TL4 **444**, TL5 **448** and TL6 **452**. Amplitude discriminator transmits the amplitude of each line connected by switch **430** to computer **410**. Computer **410** then determines which line in CCA has the greatest amplitude and transmits that information over network **102** to all participants in CCA. Computer **410** also determines which line in CCB **470** has the greatest amplitude and transmits that information over network **102** to all participants in CCB **470**. In addition, Roster Information (RI) can be transmitted over the Network. As used herein the term Roster Information (RI) means a list of speaker identifications and any other information that may be desirable to furnish a conference call participant and that can be accessed during the conference call by either a computer or a telephone display unit.

Figure 5 depicts Caller ID System 2 (CIDS2) having server computer **410** connected to amplitude discriminator **420** by line **414** and to modulator **405** by line **404**. Amplitude discriminator **420** is connected to switch **430** by line **424**. For purposes of illustration only, and not by way of limitation, switch **430** connects the same incoming lines from PSTN **480** as shown in Fig. 4 into the same conference call A (CCA) **460** and conference call B (CCB) **470**. As in

Fig. 4, Amplitude discriminator **420** transmits the amplitude of each line connected by switch **430** to computer **410**. However, in CISC2 **500**, computer **410** determines which line has the greatest amplitude and transmits this information to modulator **405** for encoding into coded signal (CS) **490** for transmission to all lines in CCA **460**. Modulator **405** is connected to switch **430** by line **406** for transmission of CS **490**. Persons skilled in the art are aware of multiple ways in which CS **490** can be transmitted digitally on a digital phone system, and also of multiple ways in which CS **490** can be encoded on an analog phone system by the use of low frequency (or other frequency, but different from those produced by human voices) tones carrying encoded identification of the speaker. For example, a system similar to that used to unlock amateur radio repeaters via low level tones would be almost inaudible on a conference call. As in Fig. 4, Roster Information (RI) can be transmitted by coded signals such as CS **490**. RI can be kept to a minimum by using shorthand designations for participants such as numbers or codes and thereby minimizing the impact of low frequency tones when such a method is used for transmission.

Figure 6A is a depiction of memory **600** located in computer **410** of Fig. 4 and Fig. 5. Memory **600** includes without limitation the storage locations for data storage **601** and program storage **602**.

Figure 6B is a depiction of data storage **601** including without limitation conference call data **604** and incoming line data **606**.

Figure 6C is a depiction of program storage **602** including without limitation data acquisition programs **608** and speaker identification programs **610**. Conference call data **604** includes without limitation a conference call list file **612** (Fig. 6D) and a plurality of conference call information files **620** (Fig. 6E).

In Fig. 6D there are two fields in conference call list file for the two conference calls designated in Figures 4 and 5 as conference calls A and B. Each conference call field in conference call list file **612** is linked by pointers to a conference call information file. For example, referring to Fig. 6E, conference call A has information file **621** and conference call B has information file **622**. Each information file has the same fields for data. By way of example, each conference call information file is shown having 8 fields as follows: (1) a conference call identification number; (2) a pass code for gaining access to the conference call; (3) the scheduled time of the conference call; (4) the scheduled duration of the conference call; (5) accounting information regarding the conference call; (6) a listing of the participants; (7) a designation of the current speaker; and (8) a current line list.

Figure 6F depicts an incoming line list file **630** with fields for identification of every incoming line. By way of example, incoming line list file **630** shows six incoming lines to correspond to the six incoming lines shown in Figures 4 and 5. However, incoming line list file **630** could consist of any number of incoming lines. Each incoming line field in incoming line list file **630** is linked by pointers to a corresponding line information file.

Fig. 6G depicts a plurality of line information files **640**. In the example, line information file **641** is linked to the field for line 1 in incoming line list file **630**, line information file **642** is linked to the field for line 2 in incoming line list file **630** and so on. Each line information file contains fields for data. By way of example, each of the plurality of line information files **640** includes: (1) identification of the line; (2) the conference call number to which the line is assigned, if applicable; (3) the average amplitude of the line; (4) identification of a speaker by voice print; and (4) a list of callers who are using the line. The fields for identification of a

speaker by voice print and the fields for callers who are using the line are linked to corresponding participant fields in a participant list file (See Fig. 6H). In incoming line list file 630, line 1 field is linked to first line information file 641, line 2 field is linked to second line information file 642, line 3 field is linked to third line information file 643, line 4 field is linked to fourth line information file 644, line 5 is linked to fifth line information file 645 and line 6 is linked to sixth line information file 646.

Figure 6H depicts participant list files 650 where there is one participant list file for each conference call. For example, in Fig. 6H first participant list file 651 may be for CCA 460 and second participant list file 652 may be for CCB 470.

Figure 6I depicts participant information files 660 containing fields for each of the participants in a conference call. In the examples of Figures 4 and 5, there are three lines in each conference call. However, there could be more than one speaker on a line because speakers could share a line using a speaker phone. Participant list file 651 depicts four data fields for four participants where there are three lines and two of the participants are sharing a line. Each of the participants identified in the fields of the participant list file 650 is linked by pointers to a corresponding participant information file 660. For example, referring to Fig. 6I, the field for participant 1 of first participant list file 651 is linked to participant information file 661. Each of the participant list files contains data fields such as (1) name of the participant; (2) the company for which the participant works; (3) the telephone number of the participant; (3) the participant's internet address; (4) the number from which the participant is calling into the conference call; (5) the participant's voice print data; and (6) other fields as desired. Participant 2 would be linked to second participant information file 662, participant 3 would be linked to third participant

information file **663** and participant 4 would be linked to fourth participant information file **664**. In the same manner, if the four participants in second participant file **652** were the same they would be linked to the same participant files as described above. If the four participants were different and referred to as participants 5 through 6, then participant 5 would be linked to fifth participant file **665**, participant 6 would be linked to sixth participant file **666**, participant 7 would be linked to seventh participant file **667** and participant 8 would be linked to eighth participant file **668**.

Figure 6J depicts conference call line lists **670**. In the example, the conference call line list corresponding to CCA in Figures 4 and 5 is conference call line list **672** containing fields for lines 3, 4 and 6. Also in the example, conference call line list **674** corresponds to CCB and would contain fields for lines 1, 2 and 5. Each of the line fields in conference call line lists **670** are linked by pointers to the corresponding file in line information files **640**.

Fig. 6K depicts links files **680** containing a links file corresponding to each conference call and containing three types of links: line links, participant links and call links. The links fields contain the pointers necessary to link the fields from one file to the corresponding fields from another file. In the example with two conference calls, CCA and CCB, first line links file **672** is for CCA and second line links file **674** is for CCB.

Figure 7 is a depiction of telephone display unit system (TDUS) **700**. TDUS **700** has PSTN **480** connected to telephone display unit (TDU) **740** by first connecting line **710**. Handset **720** is connected to first connecting line **710** by second connecting line **712** and to TDU **740** by third connecting line **722**.

TDU 740 has demodulator 750 connected to PSTN 480 by first connecting line 710, power supply 790 by fourth connecting line 752 and microprocessor 760 by fifth connecting line 754. Microprocessor 760 is directly connected to memory 770 and is connected to power supply 790 by sixth connecting line 762 and to liquid crystal display (LCD) 780 by seventh connecting line 764. LCD 780 is connected to power supply 790 by eighth connecting line 782. Power supply 790 is connected to transformer 724. Transformer 724 is connected to wall socket 728.

Figure 8 depicts computer display (CD) 800. CD 800 has computer monitor, 810, monitor screen 820, and information package display 830. Information package display 830 has toolbar 832 and various items of information display including by way of example, but not limited to, the following. First data line 834 stating that the information provided is for a conference call. Second data line 836 provides a clock showing the duration of the conference call. Third data line 838 indicates that the parties to the call will be listed below. Fourth data line 866 states the total number of parties. Fifth data line 868 states the actual time of day. Sixth data line 870 displays the date. Between third data line 838 and fourth data line 866, each of the parties to the conference call are listed along with the telephone number for each party. Check boxes 890 are provided by each name so that the speaking party can be identified. For example, in Fig. 8, check 892 appearing in check box 890 indicates that, of the four conference call participants displayed, Chris Williams 860 is the party who is speaking. Persons skilled in the art are aware of multiple methods of indicating the speaking party in addition to a check box. For example, the party's name can be highlighted, the party's name can be displayed in bold type, the party's name can flash on and off the screen and others may be used. In some cases, the amount of information to be displayed will exceed the available display area. For example, in a

large conference call there may be more participants than can be displayed on the screen. Scroll bar 880 is provided having slider 882, up button 884 and down button 886 so that the viewer can quickly scroll up or down the participant list.

Figure 9A is a depiction of telephone display unit (TDU) 900 having cabinet 910 and LCD 780. In Fig. 9A LCD 780 displays the words "Speaking" 930 and the name and telephone number of the conference caller who is speaking, Bob Jones 214-555-1234 (932), the date 936 and the time 934. In Fig. 9B, LDC 780 displays all of the participants and their telephone numbers, Bob Jones 214-555-1234 (932), Fred Smith 918-555-5678 (938), Brian Johnson 405-555-3456 (940) and Chris Williams 480-555-9876 (942) and highlights the current speaker with highlight area 948 as well as providing date 936 and time of day 934. In some cases, the amount of information to be displayed will exceed the available space on LCD 780. The information displayed may be scrolled. TDU 900 using scroll button 920 built into cabinet 910.

Figure 10 depicts roster information program (RIP) 1000. RIP 1000 begins (1002) and queries the conference caller participant for input (1004). A determination made as to whether data will be input by the operator (1006). If a determination is made that data will be input by the operator, RIP 1000 enters the data obtained by the operator (1008) and RIP 1000 goes to step 1022. If a determination is made that data will not be input by the operator, then RIP 1000 determines whether conference call numbers (CCN) have been assigned (1010). If CCNs have been assigned, the CCNs are entered (1012). If CCNs have not been assigned, then CCNs are assigned (1014). RIP 1000 displays the information (1022). A determination is made as to whether the information is complete (1022). If the information is not complete, an error message is displayed (1020) and a determination is made whether to override (1018). If a determination

is made not to override, then additional information is entered (1016) and RIP 1000 goes to step 1022. If a determination is made to override, then RIP 1000 goes to step 1026 and the information is saved (1026). If at step 1024 a determination is made that the information is complete, then the information is saved as roster information (RI) (1026). A determination is made as to whether another entry is to be made (1028). If another entry is to made, then RIP 1000 goes to step 1004. If another entry is not to be made, then RIP 1000 stops (1030).

Figure 11 depicts a flow chart of telephone display unit program (TDUP) 1100. TDUP 1100 starts (1102) and receives initial roster information (RI) (1104). RI includes as a minimum, a list of all participants in the conference call. Because the TDU display is small, the participants can be identified by numbers. Therefore, the initial RI includes the participants by name and number by which they can be referred in later transmissions. Minimizing information in the RI is also important when the RI is transmitted using low frequency inaudible or barely audible tones. A determination is made as to whether there is a speaker (1106). If a determination is made that there is not a speaker, TDUP 1100 returns to step 1104. If a determination is made that there is a speaker, then the speaker's identification is displayed (1108). A determination is made as to whether a speaker has changed (1110). If a speaker has changed TDUP 1100 goes to step 1108. If a speaker has not changed, then a determination is made as to whether changes have been made to the roster (1112). If changes have not been made, TDUP 1100 goes to step 1116. If changes have been made, the changes to the roster are displayed (1114). A determination is made as to whether the call has ended (1116). If the call has not ended, then TDUP 1100 goes to step 1106. If the call has ended, then data is cleared (1118) and the TDUP 1100 ends (1120).

Figure 12 depicts a flow chart for voice amplitude acquisition program (VAAP) 1200. VAAP begins (1202) and a determination is made as to whether there is a line with an amplitude A greater than a threshold amplitude AT (1204). AT may be zero but due to line noise and other factors, AT should be set lower than the amplitude for a speaker's voice. If no line has an amplitude greater than AT, VAAP 1200 ends (1216). If there is a line with amplitude greater than AT then the amplitude for that line is stored (1206). A determination is made as to whether there is another line with amplitude A greater than AT (1208). If there is another line with amplitude A greater than AT, then L is set equal to the next line with amplitude A greater than AT (1210) and VAAP 1200 goes to step 1206. If there is not another line with amplitude A greater than AT, then the stored amplitude samples for each line with amplitude data are averaged (1212). The average amplitude for each line with amplitude data is stored (1214) and VAAP goes to step 1204.

Figure 13 is a flow chart of voice amplitude display program (VADP) 1300. VADP 1300 begins (1302) and processes the next active conference call Z (1304). Y is set equal to the first line in the conference call (1306). AT is set equal to a value for a minimum threshold (1308). M is set equal to none (1310). VADP 1300 gets the latest average amplitude AA for line Y from the amplitude discriminator (1312). A determination is made as to whether AA is greater than AT (1314). If AA is greater than AT, then M is set equal to Y, AT is set equal to AA (1316), and VADP 1300 goes to step 1318. If AA is not equal to AT, then a determination is made as to whether there are more lines in conference call Z (1318). If there are more lines in conference call Z, then Y is set equal to the next line in conference call Z (1320) and VADP 1300 goes to step 1312. If there are no more lines in conference call Z, then line M has the maximum

amplitude and line M is the speaker's line for conference call Z (1322). If voice print capability is available, then VDAP 1300 accesses voice identification (1324). A determination is made as to whether the voice print matches the speaker for line M (1326). If the voice print matches, then speaker M is transmitted to the participants of conference call Z (1327). If the voice print does not match, then the speaker is identified from voice prints and that speaker is transmitted to participants of conference call Z (1328). Conference call Z is set to the next conference call after conference call Z (1330). A determination is made as to whether VDAP 1300 has been switched off (1332). If not, VDAP 1300 goes to step 1304. If VDAP 1300 has been switched off the program ends (1334).

Figure 14 is a flow chart of voice identification acquisition program (VIA) 1400. VIA 1400 begins (1402) and a determination is made as to whether there is a new caller (1404). If not, VIA 1400 ends (1418). If there is a new caller, a voice sample is obtained (1406). A determination is made as to whether the voice sample is sufficient (1408). If not, an additional voice sample is obtained (1410) and VIA 1400 goes to step 1408. If the voice sample is sufficient, then the voice sample is analyzed (1412). Persons skilled in the art are aware of multiple way to analyze a voice sample for use in voice identification. For example, VIA 1400 may use a real-time Fourier transform on the voice sample to discern the unique overtone patterns of the speaker's vocal cavities. Alternatively, VIA 1400 may identify phonemes in the voice sample so that unique differences can be determined by observing the curvature of spectral response maxima over time. Different speakers have different curvatures (rates, inflections) in their speech which can be utilized to identify the speech. After analyzing the voice sample, a

voice print is prepared and stored (1414). A determination is made as to whether there is another speaker (1416). If so, VIA 1400 goes to step 1406. If not, VIA 1400 ends (1418).

Figure 15 is a flow chart of the voice print identification program (VPI) 1500. VPI 1500 begins (1502). VPI determines a current voice print for each speaker (1504). A determination is made as to whether the current voice print matches a voice print in memory (1506). If there is not a match, then the current voice print is stored in memory (1508) and VPI 1500 returns to step 1506. If a match is made at step 1506, the speaker whose voice print was matched is highlighted, or otherwise designated as discussed above in Fig. 8, 9A and 9B, in the Roster Information (RI) (1510). A new RI is sent (1512). A determination is made as to whether there is a new speaker (1514). If there is a new speaker, then VPI 1500 goes to step 1504. If not, a determination is made as to whether the conference call is over (1516). If the conference call is not over, VPI 1500 goes to step 1504. If the conference call is over, then VPI 1500 ends (1518).

It will be understood from the foregoing that various modifications and changes may be made in the preferred embodiment of the present invention by those skilled in the art without departing from its true spirit. It is intended that this description is for purposes of illustration only and should not be construed in a limiting sense. The scope of the invention should be limited only by the language of the following claims.